REMARKS/ARGUMENTS

The Office Action dated May 21, 2007 has been carefully reviewed and considered. Claims 1-15, 28, 29, and 37 are currently pending and Claims 1-15, 28-29 and 37 are rejected.

Claim 3 is rejected under 35 U.S.C 112 as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regard as the invention. Claim 1-3, 5-6, 9-15, and 37 are rejected under 35 U.S.C. 102(e) as being anticipated by Anderson et al. (U.S. 2003/0084128 AL). Claim 4 and 7-8 are rejected under U.S.C. 103(a) are being un-patentable over Anderson et al. (hereinafter Anderson) (U.S. 2003/0084128 AL) in view of Zhao (U.S 6,035,404). Claim 28-29 are rejected under 35 U.S.C. 103(a) are being un-patentable over Anderson et al. (hereinafter Anderson) (U.S. 2003/0084128 AL). in view of Jackson et al. (hereinafter Jackson) (U.S 2002/0152305 AL).

In the foregoing Amendments, Claims 37 have been amended. Support for these amendments can be found in the specification and claims of the application as filed. No new matter has been added. Claims 30 has been canceled without prejudice or disclaimer of the subject matter contained therein. Claims 1-15, 28-29 and 37 are currently pending.

Applicant respectfully requests entry of the foregoing Amendments and reconsideration of the present application in light of the amendments above and the remarks below.

Objection to Specification under 35 U.S.C. 132 (a)

The Office Action objects the disclosure because there are couples of new matter throughout the Specifications. The errors in the specification have been corrected and thus, this objection is respectfully traversed.

Claim Objections

Claims 37 is objected to as allegedly being improper formalities and so forth. Claim 37 has been amended and this objection is traversed. Claim 30 is objected due to the

Election/Restriction rule and Claim 30 was restricted into Group II. Claim 30 is canceled and this objection is traversed.

Rejection Under 35 USC 102(e)

The paragraph 5 of Office Action has rejected claims 1-3, 5-6, 9-15, and 37 under Anderson. Without admitting that Anderson is prior art and reserving the right to establish that it is not prior art, Applicant respectfully traverses the rejections for the reasons below:

Regarding independent Claims 1 the Office Action asserts that Anderson teaches "a method for supporting multiple simultaneous concurrent tasks within a single web-console in a central controlled distributed scalable virtual machine ("CCDSVM") environment, said method comprises:

> (a) a user logging in from a web-console of a console host to said CCDSVM environment; (b) said user from said web-console of said console host obtaining all information of the target system within said CCDSVM environment; (c) said user from said web-console of said console host selecting a target system and initiating tasks based on said all information of said CCDSVM environment; (d) console supporting software on control management station getting and storing tasks into a user space task list, and obtaining associated locks for each task and (e) console support software distributing multiple tasks to multiple systems until all of said tasks are performed. "

The Applicant disagrees with such assertion of the Office Action for the reasons below.

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference" Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628,631,2USPQ2d 1051, 1053(Fed. Cir.1987). "The identical invention must be shown in as complete details as is contained in the claim." Richardson v Suzuki Motor Co., 869F.2d 1226,1236, 9USPQ2d 1913, 1920 (Fed. Cir. 1989) (See also MP.E.P 2131). Here and every element of Applicant's independent claims 1 and 37 are not found, either expressly or inherently, in a single prior art reference. Applicant respectfully submits that Anderson can not anticipate the presently

claimed invention because Anderson fails to disclose each and every element set forth in the claimed invention either inherently or expressly.

For one example, the Claim 1, bullet (a), (b), and (c) of present invention recites "(a) a user logging in from a web-console of a console host to said CCDSVM environment", "(b) said user from said web-console of said console host obtaining all information of the targeted systems within said CCSDVM environment", and "(c) said user from said web-console of said console host selecting a target system and initiating tasks based on said all information of said CCDSVM environment". (Emphasis added)

There are four points that Anderson has failed to suggest in his invention: First, users of Anderson have no ability to obtain all information of the targeted systems within CCDSVM environment; Second, users of Anderson have no ability of selecting a target system from multiple targeted systems and initiating tasks based on said all information of said CCDSVM environment; Third, Anderson requires both his client and local agent to periodically pull server for task information; Fourth, Anderson's system can not perform the tasks in range of present invention. Let us take one point at a time to explore why Anderson has failed to suggest each element and limit of present invention.

> First, What [0050] and [0082] of Anderson said are "[0050] Beginning with signal C1, the remote client 20 sends a task request signal to the server 10. The task request signal C1 is received by the server 10 and is then queued in the task queue 14. The local agent 40, as part of its periodic poll of server 10, polls the server 10 with signal A1. The signal A1 is received by the server 10, which then checks its task queue 14 for any task requests. The task request from signal C1 is located, so the server 10 sends or forwards the task request to the local agent 40 at signal S1.", and "[0082] In the agent processing module 340, according to one embodiment, after the local agent 40 has logged on, then the server 10 first determines whether the logon is a standard poll of the server 10 to determine whether any tasks are waiting in the task queue 14, or if the logon is a file upload. According to one embodiment, the two states are treated independently of each other--meaning if you have one state, then you do not have the other. In another

embodiment, the server 10 first receives and processes the file uploaded by the agent 40, and then checks the task queue 14 for any new tasks that need attention." (Emphasis added)

What [0032]-[0034] of Anderson have said are "[0032] A region of memory in the server 10 is reserved for a task queue 14. The task queue 14 is a special purpose memory structure for storing requests or tasks for a client/user that logs on to the server 10. These tasks and the operation of the task queue 14 will be described in further detail below. We note that the task queue 14 can be an addressable part of the database 16, or it can be a specially maintained region of memory in the server 10.", "[0033] Files 48', from the computer 44 files 48, are shown temporarily stored at the server 10. This is described in further detail below, but we note that the files 48' can be stored in the database 16, a special memory region, such as the task queue 14, or another special memory region reserved for such files.", and "[0034] The server 10 is preferably configured to be communicatively coupled with a series of clients, comprising at least a remote client 20, and a local agent 40. Connectivity can be maintained or provided through a TCP/IP, wireless access protocol (WAP), HTTP, and/or an SSL protocol, as is depicted in the connectors between the various elements depicted in FIG. 1. Typically, the server 10 connections are maintained over a network 30, for instance a wide area network (WAN), such as the Internet. If a remote client 20 is to access the server 10 through another network, such as the public switched telephone network (PSTN), or a wireless device, then an appropriate protocol is used, and the server 10, or an intermediary device, handles the translation from the needed protocol and an IP protocol. In addition to connectivity features in the communications stack 12, communication can be made using SOAP, WML, XML or VXML, HTML programming languages." (Emphasis added)

Clearly, throughout [0032]-[0034], [0050], [0082] and the entire specification of Anderson, there is no place to mention that a client can obtain all information of targeted systems.

Second, Anderson never mentions that a client at a single web-browser can select a target system from multiple systems to initiate multiple concurrent tasks based on said all information of said CCDSVM environment. Instead, Anderson suggests each client be pre-configured to access only one system, a local computer system, typically located at home as Anderson described in [0071] and [0079] as

> "[0079] In act 208, remote clients 20 can be setup. This can be done manually, by configuring the remote client in the agent software 200, or it can be done automatically. What is meant here is that remote clients 20 can be setup and managed, thereby giving a user of the local agent software 200 the ability to individually tailor access, security, or file transfer type information for particular remote clients, or globally setting such preferences, with respect to a the local computer the local agent is associated with." (Emphasis added) "[0079] FIG. 6 is a flowchart for the server software 300. We begin with act 304, where the relational database management system 16 is setup. Here, we can setup remote client 20 and local agent 40 default values, such as polling period for the local agent 40, file types for different remote client 20 types, notification messages, upload file types, and other standard information concerned with the file management. Preferably, pricing plans and other user information is stored in the database 16, so it can be setup too. In act 308, users are setup for the database 16. This can be via manual entry, or an automated process that is part of a HTML or XML based web interface on the server 10. We note that an exemplary database schema for the database 16 is depicted in FIG. 7 and described below with reference that figure." (Emphasis added)

The Anderson's [0071] and [0079] clearly reveals that his client can only access one fixed system because the client is manually or automatically configured with a fixed local agent. Also, Anderson never mentioned that a client can access multiple local systems and never mentioned that a client can have choices of selecting a system from targeted systems to run task. Besides, as mentioned previously, user of Anderson can not obtain all information of the targeted systems, therefore, it further can not initiate multiple concurrent tasks based on all obtained information of said CCDSVM environment.

Contrary to Anderson and in additional to Claim 1, (a), (b), and (c) of present invention, the first and second point of argument also have been described (in part) by present invention in the followings:

> On page 13, lines 1 – line 8 of present invention that "In one example, a user A from a web-console (9 of Fig. 2) got authenticated by console supporting software (6 of Fig. 2) such as successfully login on control management station (3 of Fig. 2). So that user A has obtained necessary information of all system units (3 of Fig. 2) and control management station (2 of Fig. 2) from console supporting software (6 of Fig 2). When user A initiates a task for a selected target system, which is either a system unit (3 of Fig. 2) or the control management station, the task information is transmitted....." (Emphasis added)
>
> On page 14, paragraphs 2 and paragraph 3 of present invention that (in part) "If the task needs to be executed on the control management station ....." and "If a task needs to be executed on a system unit ....."..

Third, both client and local agent of Anderson requires logging on, then logging out of the server and periodically pulling the server, which has been described in [0072] and [0077] of Anderson:

> "[0072] In act 212, the local agent 40 polls the server 10. This is done by logging on to the server 10, typically using a user name and password pair via a modem or a LAN connection. At act 216, a test is performed to determine whether a non-fulfilled task request exists in the task queue 14 of the server 10. If a task request does not exist, then a wait state is entered in act 220, where the local agent 40 will logoff the server 10 and then reconnect to the server 10 once the next predetermined polling period (setup in act 204) has expired. However, if a non-fulfilled task request exists in the task queue 14, then processing by the local agent software 200 continues to act 224." (Emphasis added) "[0077]Once the task request is sent to the server 10, the remote client 20 will wait for a reply from the server 10. According to one embodiment, the remote client 20 logs off of the

server 10 and the polls the server 10 periodically to determine whether the task request was completed by the local agent 40. This process is depicted in optional/alternate act 254, depicted as individual acts 274, 278, and 282. However, according to another embodiment, remote client preferences, established with the local agent 40 or the remote client 20, indicate that the server 10 must notify the remote client 20 when the task request is complete. This method is depicted a act 264, where the remote client 20 receives a notification from the server 10 that the task request from act 262 is complete.". (Emphasis added)

Contrary to such extremely in-efficient tasking executing environment of Anderson, the present invention never requires the user at the web-console to log in and then log out to periodically pull control management station and also never requires service software of server unit to log on control management station to pull periodically to find out if there is a new task. Instead, the claim 12 of present invention states that "if the target of a task is for a system unit, the console support software of control system transmits the task information to the service software of system unit." directly (Emphasis added), which is simple and straightforward. The Claim 1 of present invention with bullet (a) (b) and (c) has clearly indicated that user only need to login to CCDSVM environment, obtain all necessary information, and initiate tasks, which has been described in page 15 of Applicant's specification and in Claim 30; Also, user can run tasks according the security rules described in lines 1 – 11 on page 14 of present invention that:

"The credential of executing a specific task on a specific target system by user A is checked. In addition, the ordinary users access & operation permissions and credentials are setup by administrator with super user privileges. If user A is not permitted to perform any task on such target system or is not permitted to perform such task on any system, the task execution will be fail and user A will be notified via net (11 or 12 of Fig. 2). Otherwise, the task will be carried out by the corresponding thread on target system, which is either control management station (2 of Fig. 2) or a system unit (3 of Fig. 2). If there is need, the console

supporting software (6 of Fig. 2) will send results data back to web-console (9 of Fig. 2). When a task is either failed or succeeded, the threads of console supporting software (6 of Fig. 2) will release the locks acquired for this task." (Emphasis added)

Fourth, the Anderson's invention can not perform the tasks in claim 1 of present inventions such as described (in part) in page 15 of specification that:

"a) Move or transmit data such as a multiple gigabytes of file or other data in any form from any point or any system to another point or system within CCDSVM (Fig. 2).

b) Configure, partition and design entire storage system (raid/disk) within CCDSVM

....

d) Monitor and display activities and status for network, storage, CPU, processes and threads...".

These tasks never being disclosed by Anderson and can not be performed by Anderson's invention, which will be further discussed in next examples.

In summary, based on four points discussed, there are significant different behavior in executing tasks between local agent of Anderson and service software of system unit in CCDSVM of present invention, and between remote client of Anderson and user at web-console of present invention have proved enough that Anderson failed to suggest and anticipate the elements of present invention either expressly or inherently. Thus for these reasons alone, the Applicant submit that the Claim 1 is patentable over Anderson's invention.

For another example, the Claim 1, bullet (d) and (e) of present invention recites "(d) console support software on control management station getting and storing tasks into a user space task list, and obtaining associated locks for each task. (e) console supporting software distributing multiple tasks to multiple systems until all of said tasks are performed".

Although the Office Action asserts that [0032], [0040], [0058], [0059] of Anderson teach Claim 1 bullet (d) of present invention and, [0082]-[0083] of Anderson teaches Claim 1 bullet (e) of present invention, the Applicant disagree with such assertion due to two facts:

First, Anderson has failed to suggest the limits of tasks in claim 1of present invention, which has been discussed in first example. Second, more significantly, [0082]-[0083] of Anderson failed to suggest for supporting user to perform concurrent simultaneous tasks from a single conventional web-browser.

> Again, Anderson states that "[0082] In the agent processing module 340, according to one embodiment, after the local agent 40 has logged on, then the server 10 first determines whether the logon is a standard poll of the server 10 to determine whether any tasks are waiting in the task queue 14, or if the logon is a file upload. According to one embodiment, the two states are treated independently of each other--meaning if you have one state, then you do not have the other. In another embodiment, the server 10 first receives and processes the file uploaded by the agent 40, and then checks the task queue 14 for any new tasks that need attention.", and "[0083] according to another embodiment, in act 344, the poll is received. Task requests in a task list corresponding to the local agent 20, are queried or looked up in the task queue 14 in act 348. In act 352, a test is performed to determine whether there are any outstanding tasks in the task queue 14. If there are no outstanding task requests in the task queue 14, then the local agent 40 is informed of such and logged out (act 380). However, if there are task requests in the task queue 14, then processing continues to act 360."

Obviously, the [0082]-[0083] of Anderson only describe how a user task, which might be initiated from a web-browser on client system and can be executed after the task being queued in task queue. However, the Applicant submit that the method described in [0082]-[0083] does not solve the problem of user's web-browser could be freezing during initiating and executing tasks if these takes longer time to be run. As a result, there will be no other tasks can be initiated from the same browser concurrently. This issue has been raised and described in section of "Brief Description of This Invention" on paragraph 1 of page 6 of present invention. On the other hand, Anderson has failed to suggest resolving

such problem. Further, in additional to [0082]-[0083] of Anderson, there are also several key descriptions and figures of Anderson revealing the same problem.

First, look at Anderson's client method of [0077] "Once the task request is sent to the server, the remote client will wait for a reply from server. According to one embodiment, the remote client logs off the server and polls the server periodically to determine whether the task request was completed by the local agent". (Emphasis added)

Second, look at Anderson's Fig. 5, with optional/alternate act 254, where user enters waiting and polling state. Third, let us look at same Fig. 5 of Anderson between act 262 and 264, where a user also has to wait for receiving a notification or signal, or wait in act 250.

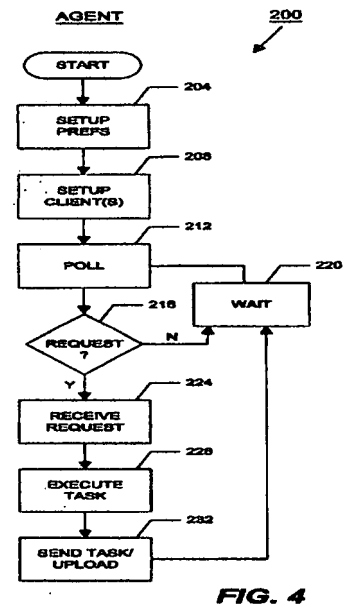

FIG. 4

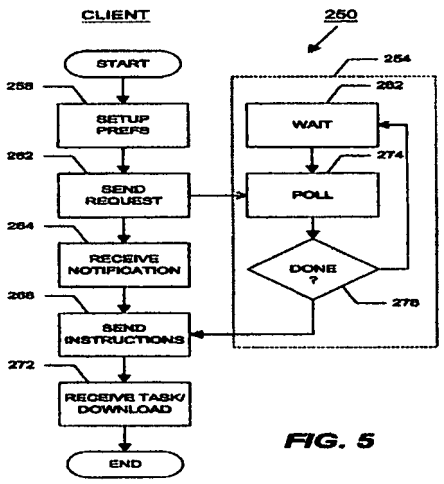

FIG. 5

Third, [0098] of Anderson has mentioned "The remote client 20 can then select a particular file from the information stored in the browse information table 728 and create a new task request to send to the server

10. The new task request will be stored in the task tables, and the local agent 40 will poll the server 10, recognizing the new task request in the task queue 14. The local agent 40 will receive the task request from the server 10 and process the task. The particular file will, in turn, be uploaded to the server 10, where it will be stored in the stored file table 732." (Emphasis added)

A simple fact is that if a user initiating a task and has to wait a long time for its completion or just log off then pull the server, such user never can initiate other concurrent tasks from same web-browser UI. This is definitely impossible either on browser-based UI or command line UI or native window-based UI. In either cases of waiting or logging off server, the user of Anderson will have no chance to use the same web-browser on their hand to perform other tasks concurrently.

In the "Brief Description of This Invention" section of current invention, this kind of problems has been fully described. Let us analysis a detailed example with Fig. 2 of Anderson to see why Anderson can not support a user from a single web-browser to initiating multiple concurrent tasks:

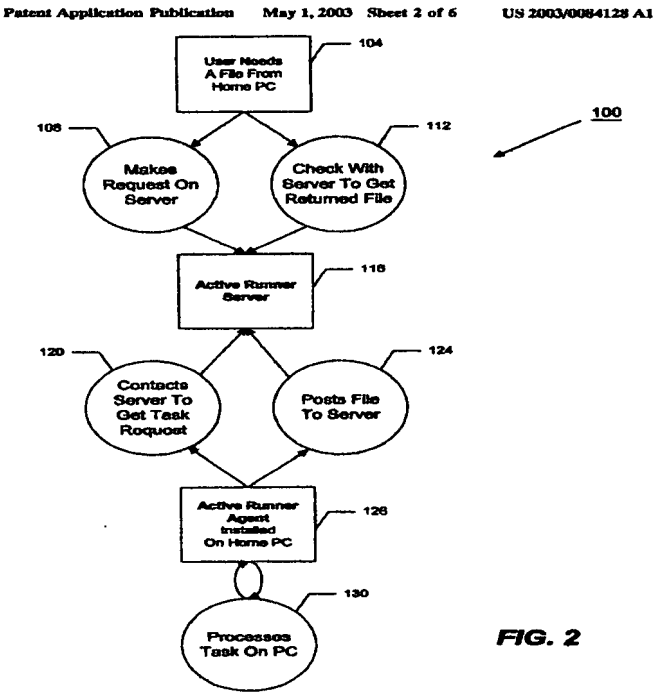

FIG. 2

If a client perform a task of accessing & downloading a 2-GB file from web-browser with act 108 to wait & pull the server for notification of task completion with act 112, the server then queues the task and will check if the local agent has finished the task with act 115, and the local agent periodically logs onto the server and pulls & checks to see if there is a task with act 120. If there is a task, then execute task (upload 2-GB file) on local computer and return results (deposit 2-GB file on server) with act 124. Remote client finally get result (2GB file) after detecting the notification (of file upload completed). How long the client at his browser has to wait from initiating such task to finally get file?

With Anderson's invention, the 2-GB file travels through two major network segments: one from local computer to server and the second from server to client computer. If Anderson's client is remotely on the Internet and if client's corresponding local computer connection speed is 6-mbps (the best DSL ISP provided in US, which is about 750-bytes/sec in theory, actual upload speed is around 300bytes/second), user must wait for 46 minutes in theory and actually may wait for 2 hours to finish upload file from local computer to server; If the connection speed is 45-mbps (DS3/T3), user must wait for 5.2 minutes in theory to finish the upload; If the connection speed is 155 mbps (OC-3), user will wait about 1 minute and 48 seconds in theory to finish upload. With fastest connection of 155mbps, user has to wait 1 minute and 48 seconds at a web-UI screen in order to get a notification that the 2-GB file is uploaded on server and is available for download. Even if the connection enhanced to 1024mbps (Giga bits /second) in the future, user still need to wait for 16 more seconds in theory, which is too long to effectively initiate another simultaneous concurrent task within same web-browser.

During waiting time, the user will experience the web-browser UI frozen; user can not type anything; and consequently can not initiate other concurrent tasks from same web-browser UI (not the native Windows UI). With Anderson, worsted things is that user of Anderson need to have another same amount of time to perform download from server to client site. With Anderson's invention if user initiates time consuming tasks such as configure multi-tera-bytes of raid storage system, create or transfer multi-GB of data file, the similar web-screen freezing will occur. Obviously, Anderson has failed to suggest

concurrent multi-tasks support within a single web-browser UI because his invention only can work for very simple and non time consuming tasks, which do not require to be run concurrently. On the other hand, the novelty of an invention shall works with its defined scope for all cases.

The Applicant's invention can provide multiple users and each user from a single web-browser can initiate multiple concurrent tasks on systems in CCDSVM environment due to two facts: one is that the present invention providing user space task list to avoid multiple users' tasks being blocked or waited, and to protect data integrity of each tasks. Another is that the present invention has provided a way to let user do not have to wait for the tasks being initiated within same web-browser UI, therefore, user can start initiating another task from same web-browser UI immediate. This has been mentioned in lines 1 - 11 on page 14 of present invention that

> "The credential of executing a specific task on a specific target system by user A is checked. In addition, the ordinary users access & operation permissions and credentials are setup by administrator with super user privileges. If user A is not permitted to perform any task on such target system or is not permitted to perform such task on any system, the task execution will be fail and user A will be notified via net (11 or 12 of Fig. 2). Otherwise, the task will be carried out by the corresponding thread on target system, which is either control management station (2 of Fig. 2) or a system unit (3 of Fig. 2). If there is needs, the console supporting software (6 of Fig. 2) will send results data back to web-console (9 of Fig. 2). When a task is either failed or succeeded, the threads of console supporting software (6 of Fig. 2) will release the locks acquired for this task."

We shall notice that with the Applicant's invention, only "If there is needs, the console supporting software (6 of Fig. 2) will send results data back to web-console", in other words, normally the user does not have to wait for the results or notification at their web browser UI. This adequately ensures that user can initiate other simultaneous concurrent tasks immediately from same single web UI of present invention. Clearly, present invention has overcome a major hurdle that Anderson has encountered and failed to suggest. Because

Anderson's method has failed to support user initiating multiple concurrent tasks within same web-browser, Anderson has failed to suggest the elements and the limits of present invention. Consequently, the independent Claim 1 is patentable over Anderson and the rejection of Office Action on Claim 1 of present invention must be withdrawn.

Regarding independent claim 37, the Office Action has rejected claim 37 under 35 U.S.C. 102, alleging that "Anderson teaches a method for supporting multiple simultaneous concurrent tasks with a single web-consoles comprises:

> Providing a group of computer systems having at least one control system and zero or more server systems and connected together through network media, wherein said group of computer systems are controlled, operated, and managed by said control system with a set of software modules running on either control system or on each server systems in said group of systems (Fig. 1, item 4, 10, 20, 40,44, 48, etc.);
>
> Providing multiple users login concurrently each from web-browser of client system into said control system ([0037],[0072]);
>
> Providing said users each from a single web-browser on client system to obtain information relating to system configuration and resources information of control system and server systems within said group of systems([0050],[0082]);
>
> Providing said users each from a single web-browser of client systems to select said target systems, which is either a said control system or server systems, and to initiate multiple simultaneous concurrent tasks over the said configuration and resources information on selected target systems ([0082]-[0083]);
>
> Providing said web console supporting software on control management station gets and stores tasks from each users on client systems into an user space task list, and also obtains the associated locks for each tasks ([0032],[0040],[0058],[0059], Fig. 1, item 14); and
>
> Executing tasks arranged by said console supporting software, which run on the target systems until the tasks got finished ([0032],[0040],[0058],[0059]).

Anderson describes how a single client or computer interacts with a web server, it also describes that there are multiple computers or clients on the network that can simultaneously access the internet web sever ([0029]-[0030]. etc.)."

In support of such assertion, the Office Action cites Anderson ([0037],[0072], ([0050],[0082]), ([0082]-[0083]), and ([0032],[0040],[0058],[0059]). However, all of these have been discussed in previous arguments for independent Claim 1 and have been proved that Anderson failed to suggest a user from a single web-browser UI to initiate multiple simultaneous concurrent tasks on targeted systems of either control system or system units.

In addition, the citation from [0029-[0030] of Anderson also can not support the assertion of Office Action, where Anderson states that

"[0029] Turning first to the system configuration depicted under callout 4, we begin with a server 10. The server 10 is typically a web server and can run on a commercially available computer, such as a Sun Microsystems Enterprise Server.TM., available from Sun Microsystems in Mountain View, Calif., or a Dell.TM. or Gateway.TM. branded internet or application server. Such a system will include one or more microprocessors, a volatile memory area, a persistent memory area, and one or more mass storage devices. One or more sections of computer program code, or software, either in a compiled or an interpreted form, will run, for instance, in one of the memory areas, to cause the microprocessor(s) to perform the sequences of operations and techniques described below. [0030] The server 10 should include a communications software stack 12, such as an IP (internet protocol) stack, and should be able to handle hypertext transfer protocol (HTTP) requests, secure socket layer (SSL) transactions, as well as a form of a standard generalized markup language (SGML), such as extensible markup language (XML), wireless markup language (WML), and optionally voice extensible markup language (VXML). Preferably, the variant of XML employed on the server is Microsoft's SOAP.TM. (Simple Object Access Protocol), although Java.TM. or X Windows.TM. could alternatively be employed. Hypertext markup language (HTML) files are preferably included on the server 10. The communications software stack 12 and the programming languages mentioned

above are generally known in the art of network communications and interface design and are widely available."

A simple fact is that none of web server, web-protocols (SOAP, HTTP, IP stack) and programming languages such as XML, WML, HTML have solved the issues of multiple concurrent tasks support within a single web-browser. Respectfully, these citations has failed to support the assertion of the Office Action. Accordingly, Anderson does not and can not teach or suggest Applicant's independent claim 37 either. Consequently, the Office Action's rejection of independent claim 37 should be withdrawn.

In view of the above, Applicant's independent claim 1 and 37 distinguish patentability over the Anderson and should be allowed. If the independent claims are valid, the claims that depend from the independent claims should also be valid as matter of law. See Jenric/Pentron, Inc. v. Dillon Co., 205 F. 3d 13377, 1382 (Fed. Cir.2000). Since claims 2, 3, 4, 5-6, 7-8, and 9-15 are depend on allowable independent Claim 1, Applicant respectfully submit that claims 2, 3, 4, 5-6, 7-8, and 9-15 should also be patentable as matter of law.

Rejections Under 35 USC 103(a)

In Paragraphs 23, Office Action has rejected claim 4 and 7-8 under 35 U.S.C. 103(a) as being un-patentable over Anderson (US 2003/0084128 AL) in view of Zhao (US 6,035,404). Applicant respectfully traverses the rejection and disagrees with Office Action assertion of claims 4, and 7-8 in view of following remarks.

Pursuant to MPEP § 2143.03:

> 2143.03 All Claim Limitations Must Be Taught or Suggested
> To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. In re Royka, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." In re Wilson, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending

therefrom is nonobvious. In re Fine, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) (emphasis added).

Thus, under MPEP § 2143.03 and well-settled case law, where the rejection is based on a combination of references, the claim limitations must be taught or suggested by at least one of the references. Consequently, Applicant can show non-obviousness by attacking references individually where the rejection is based on a combination of references.

Applicant's independent claims are 1 and 37. None of these claims have been rejected on the ground of obviousness by the Office Action under 35 U.S.C. 103(a). Additionally, for the same reason discussed before, Anderson failed to suggest on independent claim 1 and 37 of Applicant. It shall be noticed that Claim 4, and 7-8 are dependent claims for Independent Claim 1. Consequently, the Office Action has failed to establish prima facie obviousness of dependent claim 4 and 7-8 because the elements and limits of claim 1 are not taught or suggested by Anderson. Based on this alone, the rejection on Claim 4, and 7-8 of Office Action must be withdrawn.

In addition, Applicant recites the claim 7 and 8 of present invention that

"7. (Currently Amended) The method of claim 1, wherein, step (c) further includes:

initiating multiple simultaneous concurrent tasks for a target system or for several different target systems from a single web-console of a console host in response to information obtained by said console support software; and [[.]]

sending [[The]] task information from said web-console of console host to said console support software modules of said control management station." , and "8. (Currently Amended) The method of claims 1, wherein, step (c) further includes:

[[Based]] based on information obtained from said console support software of said control management station, multiple users on multiple console hosts, each of them from a single web-console of multiple console hosts, initiating multiple simultaneous concurrent tasks for a target system or for several target systems[[.]];

[[The]] the target system [[could]] being any of system units or a control management station[[.]]; and

[[The]] task information is send from said web-console of console host to said console support software modules of said control management station."

The Office Action asserts (in part) that "As to claim 4, Anderson's invention can having multiple users logging in with their respective username/password on various console hosts. Anderson is silent on whether they can login concurrently. Zhao teaches multiple users logging in said console host through multiple web-consoles of multiple different console host concurrently (col. 7, lines 15-32, see Abstract) .... As to claim 7-8, they are rejected for similar reasons to the rejection of claim 4....".

What Zhao col.7, line 15-32 says are "If the user is found to be a concurrent user in block 62, the system next determines from the mask data if the maximum number of simultaneous logins are existing at the present time (block 78). If not, 1 is added to the user login map (block 80), the login is permitted , and flow control proceeds to block 68 for additional processing. If the maximum number of logins already exists, then the state lookup table (SLT) is consulted to determine which session for the same concurrent user IUID has been established for the longest time. That session is then given a termination or "time out" time (block 82) and that information is entered into the SLT. Flow control then proceeds to block 68 for additional processing. As in the case of the single user forced login situation, the concurrent user here may be advised of the time left for the current session to be timed out, thereby indicating to the logging in user the maximum time he needs to wait until he can get access to the system.".

The Applicant disagrees with the assertion made by Office Action due to two reasons: First, from col.7, line 15-32 of Zhao's, there is no evidence shows that the control "mask" of Zhao can control login user to select a target system from multiple targeted systems on stateless network but rather control & limit login user to access a restricted system on stateless network (also see first sentence in Abstract of Zhao that "System and method for managing user logins to a restricted computer service over a stateless network", and see

claim 1- claim 8 of Zhao). Opposite to Anderson, Applicant's invention control login user to access one or more targeted systems in stateless network. Second, as quoted and explored before that with Anderson, login users cannot run tasks on control management station and cannot select and run tasks on any targeted systems but only on a pre-configured system such as local computer of home system while present invention has no such limits, which has been described in claim 7-8 of present invention and demonstrated in following:

1) in paragraph 2 of page 6 of Applicant's specification
2) in paragraph 1 of page 13 of Applicant, paragraph 1 of page 14 of present invent
3) in paragraph 2 & 3 on page 14 of present invention
4) in Fig. 1 and Fig 2 of present invention.

For example, the paragraph 2 of page 6 in specification of present invention states "With this invention, the multi-tasks support on the web-console in a simple environment (Fig 1) has been viewed as a special case of such support in a CCDSVM environment (Fig. 2). The CCDSVM (Fig. 2) will be degenerated into a simple server environment (Fig. 1) if multiple system units (3 of Fig. 2) do not present".

The paragraph 1 of page 13 in specification of present invention states "So that user A has obtained necessary information [[of]] about all system units (3 of Fig. 2) and control management station (2 of Fig. 2) from console supporting software (6 of Fig 2). When user A [[initiated]] initiates a task for a selected target system, which is either a system unit (3 of Fig. 2) or the control management station (2 of Fig. 2),"

The page 14 of Applicant's specification states that " If user A is not permitted to perform any task on such target system or is not permitted to perform such task on any system, the task execution will [[be]] fail and user A will be notified via net (11 or 12 of Fig. 2). Otherwise, the task will be carried out by the corresponding thread on the target system, which is either a control management station (2 of Fig. 2) or a system unit (3 of Fig. 2)".

Clearly, the combined references from Anderson and Zhao still failed to suggest how to control login user run tasks over control system or over one or multiple system units by

providing user to select a target system from targeted systems. Again, the Office Action has failed to establish prima facie of obviousness for claim 4 and claim 7 - 8 because the limitation of claim 4, claim 7- 8 also can not be taught or suggested by combination of Anderson and Zhao. Consequently, the Office Action's rejection of dependent claims 4, and claim 7-8 based on obviousness must be withdrawn.

In Paragraphs 26, Office Action also has rejected Claim 28-29 under 35 U.S.C. 103(a) as being un-patentable over Anderson (US 2003/0084128 AL) in view of Jackson (US 2002/0152305). The Office Action further asserts that (in part) ".... Anderson is silent in having a first level of security authentication for the control management station and a second level of security for the system units. However, Jackson teaches having and specifying security levels (with username/passwords) fro a plurality of processing engines with the capability of define logical volumes such as size....".

Applicant respectfully traverses the rejection and disagrees with Office Action assertion on Claim 28-29 with following view:

> What [0400] of Jackson described is "[0400] In addition to selectably interconnecting particular first processing engine/s 2100 to particular second processing engine/s 2120 using one or more distributed interconnect/s 2110, it is also possible to manage operations of an application specific buffer/cache 2122 and/or a file system specific logical volume manager 2124 over one or more distributed interconnect/s 2110 via a separate reserved or dedicated communication path across the distributed interconnect 2110. In this regard, operations of a buffer/cache 2122 that may be so managed include, but are not limited to, configuration, gathering performance characteristics or data (e.g., to verify proper functioning), specifying security levels/gates (e.g., passwords, etc.). Operations of a logical volume manager 2124 that may be so managed include, but are not limited to, configuration (e.g., defining logical volumes and/or characteristics of logical volumes, such as defining number of RAID mirrors and size), loading content on to the logical volume manager (e.g., without interfering with user-access to data), etc. Advantageously, this embodiment may be used to provide a separate or reserved communication path

for buffer/cache and/or logical volume manager management communication between a first processing engine 2100 and a second processing engine 2120 (e.g., inter-processor command communication between an application processing engine and a storage processing engine) that does not interfere with or reduce bandwidth for data/information (e.g., content) exchange between the first processing engine 2100 and the second processing engine 2120." (Emphasis added).

Unfortunately, the [0400] and rest of specification of Jackson only mentioned security levels with username/password, but never suggest or explain in details on what are the other security levels are, which has been suggested in present invention with the first level and the second level security in claim 28-29. Again let us recall that "The identical invention must be shown in as complete details as is contained in the claim." Richardson v Suzuki Motor Co., 869F.2d 1226,1236, 9USPQ2d 1913, 1920 (Fed. Cir. 1989) (See also MP.E.P 2131). Therefore, both Anderson and Jackson failed to suggest the limits of Claim 28-29. Further, both Anderson and Jackson failed to suggest the tasks of claim 28-29 of present invention, which can be executed at first level and second level of present invention as disclosed in first paragraph of page 15. Again, the Office Action has failed to establish prima facie of obviousness for Claim 28-29 because even combined Anderson and Jackson still failed to suggest the detailed first level and second level security of present invention. Therefore, the rejection on claim 28-29 based on obviousness also must be withdrawn.

Conclusion

Based on the foregoing remarks, Applicant clearly demonstrated that the present invention provides user from single web-browser to initiate multiple simultaneous tasks for systems in CCDSVM environment while combined Anderson, Zhao, and Jackson failed to suggest or disclose such technology and methods. Therefore, Applicant believes that the

rejections in the Office Action of May 21, 2007 are fully overcome, and that the application is in condition for allowance. The issuance of a formal Notice of Allowance at an early date is requested.

Applicant thanks the Examiner for carefully examining the present application. If there are any questions, the Examiner is invited to contact Applicant at (408) 813-0536.

Date:_____ By:_____

Sheng (Ted) Tai Tsao

Sheng (Ted) Tai Tsao
2979 Heidi Drive,
San Jose, CA 95132
Tel. 408-813-0536